

IN THE CLAIMS:

1. (Currently amended) A method of updating an object association between a source object and a target object, comprising:

updating a target value holder of the target object with a new source to identify the source object, ~~the target value holder comprising a value holder that the target object points to wherein the target value holder is an object that wrappers an instance of the source object in order to implement a proxy pattern;~~ and

updating a source value holder of the source object with a new target to identify the target object, ~~the source value holder comprising a value holder that the source object points to;~~

wherein [[a]] the source value holder is an object that wrappers [[a]] an instance of the target object or source object in order to implement a proxy pattern.

2. (Currently amended) The method of claim 1, wherein if the object association is a one to one object association, the step of updating the target value holder includes:

setting the new target in the target value holder from the source object, and

removing [[the]] a target from [[the]] an old target value holder in the source object, if the ~~collection~~ old target value holder already contains the target; and the step of updating the source value holder includes:

removing [[the]] a source from [[the]] an old source value holder in the target object, if the ~~collection~~ old source value holder already contains the source; and

setting the new source in the source value holder from the target object.

3. (Currently amended) The method of claim 1, wherein if the object association is a one to many object association, the step of updating the target value holder includes:

adding the new target to the target value holder ~~collection~~ from the source object, if the ~~collection~~ target value holder does not already contain the new target;

and the step of updating the source value holder includes:

removing ~~[[the]]~~ a source from ~~[[the]]~~ an old source value holder in the target object, if the ~~collection~~ old source value holder already contains the source; and

setting the new source in the source value holder from the target object.

4. (Currently amended) The method in claim 1, wherein if the object association is a many to one object association, the step of updating the target value holder includes:

setting the new target in the target value holder from the source object, and removing ~~[[the]]~~ a target from ~~[[the]]~~ an old target value holder ~~collection~~ in the source, if the ~~collection~~ old target value holder already contains the target;

and the step of updating the source value holder includes:

adding the new source to the source value holder ~~collection~~ from the target, if the ~~collection~~ source value holder does not already contain the new source.

5. (Currently amended) The method in claim 1, wherein if the object association is a many to many object association, the step of updating the target value holder includes:

adding the new target to the target value holder ~~collection~~ from the source object, if the ~~collection~~ target value holder does not already contain the new target;

and the step of updating the source value holder includes:

adding the new source to the source value holder ~~collection~~ from the target object, if the ~~collection~~ source value holder does not already contain the new source.

6. (Currently amended) The method in claim 1, wherein the target value holder includes a method for setting a source ~~and target objects~~ object to be wrapped by the target value holder, and a method for returning ~~source and a target objects~~ object wrapped by the source value holder; and

wherein the source value holder includes a method for setting a target object to be wrapped by the source value holder, and a method for returning a source object wrapped by the target value holder.

7. (Currently amended) The method in claim 1, wherein the steps of updating a target object is and updating a source are performed using a source value holder implemented in Java, C++, Smalltalk, Eiffel, or other object oriented language; and wherein the steps of updating a source object is performed using a target value holder implemented in Java, C++, Smalltalk, Eiffel, or other object oriented language.

8. (Currently amended) The method in claim 1, wherein the target value holder receives as input parameters one or more of:

- a value holder that an old source currently points to;
- a value holder that an old target currently points to;
- a value holder that a source currently points to;
- a value holder that a target currently points to;
- a source value;
- a target value;
- a source cardinality; and
- a target cardinality; and

wherein the source value holder receives as input parameters one or more of:

- a value holder that an old source currently points to;
- a value holder that an old target currently points to;
- a value holder that a source currently points to;
- a value holder that a target currently points to;
- a source value;
- a target value;
- a source cardinality; and
- a target cardinality.

9. (Currently amended) A method of deleting an object association between a source object and a target object, comprising:
- updating a source value holder of the source object to identify the target object, ~~the source value holder comprising a value holder that the source object points to~~ wherein the source value holder is an object that wrappers an instance of the target object in order to implement a proxy pattern; and
 - updating a target value holder of the target object to identify the source object, ~~the target value holder comprising a value holder that the target object points to;~~
 - wherein ~~[[a]] the target~~ value holder is an object that wrappers ~~[[a]] an instance of the target object or source object~~ in order to implement a proxy pattern.
10. (Currently amended) The method of claim 9, wherein if the object association is a one to many object association, the step of updating the source value holder includes:
- setting ~~[[the]]~~ a value in the source value holder to null;
 - and the step of updating the target value holder includes:
 - removing ~~[[the]]~~ a target from the target value holder ~~collection~~ in the source object, if the ~~collection~~ target value holder already contains the target.
11. (Currently amended) The method of claim 9, wherein if the object association is a many to many object association, the step of updating the source value holder includes:
- removing ~~[[the]]~~ a source from the source value holder ~~collection~~ in the target object, if the ~~collection~~ source value holder already contains the source;
 - and the step of updating the target value holder includes:
 - removing ~~[[the]]~~ a target from the target value holder ~~collection~~ in the source object, if the ~~collection~~ target value holder already contains the target.
12. (Currently amended) The method of claim 9, wherein the target value holder includes a method for setting a source ~~and target objects~~ object to be wrapped by the target value holder, and a method for returning ~~source and a target objects~~ object wrapped by the source value holder; and

wherein the source value holder includes a method for setting a target object to be wrapped by the source value holder, and a method for returning a source object wrapped by the target value holder.

13. (Currently amended) The method of claim 9, wherein the steps of updating a target object is ~~and updating a source~~ are performed using a source value holder implemented in Java, C++, Smalltalk, Eiffel, or other object oriented language; and wherein the steps of updating a source object is performed using a target value holder implemented in Java, C++, Smalltalk, Eiffel, or other object oriented language.

14. (Currently amended) The method of claim 9, wherein the target value holder receives as input parameters one or more of:

- a value holder that a source currently points to;
- a value holder that a target currently points to;
- a source value;
- a target value;
- a source cardinality; and
- a target cardinality; and

wherein the source value holder receives as input parameters one or more of:

- a value holder that a source currently points to;
- a value holder that a target currently points to;
- a source value;
- a target value;
- a source cardinality; and
- a target cardinality.

15. (Currently amended) A computer program product in a computer readable medium for updating an object association between a source object and a target object, comprising:

first instructions for updating a target value holder of the ~~source~~ target object with a new source to identify the source object, ~~the target value holder comprising a value holder that the target object points to wherein the target value holder is an object that~~ wrappers an instance of the source object in order to implement a proxy pattern; and

second instructions for updating a source value holder of the source object with a new target to identify the target object, ~~the source value holder comprising a value holder that the source object points to;~~ wherein [[a]] the source value holder is an object that wrappers [[a]] an instance of the target object or source object in order to implement a proxy pattern.

16. (Currently amended) The computer program product of claim 15, wherein if the object association is a one to one object association, the first instructions for updating the target value holder includes:

setting the new target in the target value holder from the source object, and removing ~~[[the]]~~ a target from ~~[[the]]~~ an old target value holder in the source object, if the ~~collection~~ old target value holder already contains the target;

and the second instructions for updating the source value holder includes:

removing ~~[[the]]~~ a source from ~~[[the]]~~ an old source value holder in the target object, if the ~~collection~~ old source value holder already contains the source; and

setting the new source in the source value holder from the target object.

17. (Currently amended) The computer program product of claim 15, wherein if the object association is a one to many object association, the first instructions for updating the target value holder includes:

adding the new target to the target value holder ~~collection~~ from the source object, if the ~~collection~~ target value holder does not already contain the new target;

and the second instructions for updating the source value holder includes:

removing ~~[[the]]~~ a source from ~~[[the]]~~ an old source value holder in the target object, if the ~~collection~~ old source value holder already contains the source;
and

setting the new source in the source value holder from the target object.

18. (Currently amended) The computer program product in claim 15, wherein if the object association is a many to one object association, the first instructions for updating the target value holder includes:

setting the new target in the target value holder from the source object, and
removing ~~[[the]]~~ a target from ~~[[the]]~~ an old target value holder ~~collection~~ in the source, if the ~~collection~~ old target value holder already contains the target;

and the second instructions for updating the source value holder includes:

adding the new source to the source value holder ~~collection~~ from the target, if the ~~collection~~ source value holder does not already contain the new source.

19. (Currently amended) The computer program product in claim 15, wherein if the object association is a many to many object association, the first instructions for updating the target value holder includes:

adding the new target to the target value holder ~~collection~~ from the source object,
if the target value holder does not already contain the new target;

and the second instructions for updating the source value holder includes:

adding the new source to the source value holder ~~collection~~ from the target object, if the ~~collection~~ source value holder does not already contain the new source.

20. (Currently amended) The computer program product of claim 15, wherein the target value holder includes instructions for setting a source ~~and target objects~~ object to be wrapped by the target value holder, and instructions for returning ~~source and a target objects~~ object wrapped by the source value holder; and

wherein the source value holder includes instructions for setting a target object to be wrapped by the source value holder, and instructions for returning a source object wrapped by the target value holder.

21. (Currently amended) The computer program product in claim 15, wherein the first instructions for updating a target object is ~~and the second instructions for updating a source~~ are performed using a source value holder implemented in Java, C++, Smalltalk, Eiffel, or other object oriented language; and

wherein the second instructions for updating a source object is performed using a target value holder implemented in Java, C++, Smalltalk, Eiffel, or other object oriented language.

22. (Currently amended) The computer program product in claim 15, wherein the target value holder receives as input parameters one or more of:

- a value holder that an old source currently points to;
- a value holder that an old target currently points to;
- a value holder that a source currently points to;
- a value holder that a target currently points to;
- a source value;
- a target value;
- a source cardinality; and
- a target cardinality; and

wherein the source value holder receives as input parameters one or more of:

- a value holder that an old source currently points to;
- a value holder that an old target currently points to;
- a value holder that a source currently points to;
- a value holder that a target currently points to;
- a source value;
- a target value;
- a source cardinality; and
- a target cardinality.

23. (Currently amended) A computer program product in a computer readable medium for deleting an object association between a source object and a target object, comprising:

first instructions for updating a source value holder of the source object to identify the target object, ~~the source value holder comprising a value holder that the source object points to~~ wherein the source value holder is an object that wrappers an instance of the target object in order to implement a proxy pattern; and

second instructions for updating a target value holder of the target object to identify the source object, ~~the target value holder comprising a value holder that the target object points to;~~

wherein ~~[[a]]~~ the target value holder is an object that wrappers ~~[[a]]~~ an instance of the target object or source object in order to implement a proxy pattern.

24. (Currently amended) The computer program product of claim 23, wherein if the object association is a one to many object association, the first instructions for updating the source value holder includes:

setting ~~[[the]]~~ a value in the source value holder to null;

and the second instructions for updating the target value holder includes:

removing ~~[[the]]~~ a target from the target value holder ~~collection~~ in the source object, if the ~~collection~~ target value holder already contains the target.

25. (Currently amended) The computer program product of claim 23, wherein if the object association is a many to many object association, the first instructions for updating the source value holder includes:

removing ~~[[the]]~~ a source from the source value holder ~~collection~~ in the target, if the ~~collection~~ source value holder already contains the source;

and the second instructions for updating the target value holder includes:

removing ~~[[the]]~~ a target from the target value holder ~~collection~~ in the source object, if the ~~collection~~ target value holder already contains the target.

26. (Currently amended) The computer program product of claim 23, wherein the target value holder includes instructions for setting a source ~~and target objects~~ to be wrapped by the target value holder, and instructions for returning ~~source and a target objects~~ object wrapped by the source value holder; and
wherein the source value holder includes instructions for setting a target object to be wrapped by the source value holder, and instructions for returning a source object wrapped by the target value holder.

27. (Currently amended) The computer program product of claim 23, wherein the first instructions for updating ~~an old a source object is and the second instructions for updating a target~~ are performed using a target value holder implemented in Java, C++, Smalltalk, Eiffel, or other object oriented language; and
wherein the second instructions for updating a target object is performed using a source value holder implemented in Java, C++, Smalltalk, Eiffel, or other object oriented language.

28. (Currently amended) The computer program product of claim 23, wherein the target value holder receives as input parameters one or more of:
a value holder that a source currently points to;
a value holder that a target currently points to;
a source value;
a target value;
a source cardinality; and
a target cardinality; and

wherein the source value holder receives as input parameters one or more of:
a value holder that a source currently points to;
a value holder that a target currently points to;
a source value;
a target value;
a source cardinality; and
a target cardinality.

29. (Currently amended) An apparatus for updating an object association between a source object and a target object, comprising:

means for updating a target value holder of the target object with a new source to identify the source object, ~~the target value holder comprising a value holder that the target object points to~~ wherein the target value holder is an object that wrappers an instance of the source object in order to implement a proxy pattern; and

means for updating a source value holder of the source object with a new target to identify the target object, ~~the source value holder comprising a value holder that the source object points to;~~ wherein [[a]] the source value holder is an object that wrappers [[a]] an instance of the target object or source object in order to implement a proxy pattern.

30. (Currently amended) An apparatus for deleting an object association between a source object and a target object, comprising:

means for updating a source value holder of the source object to identify the target object, ~~the source value holder comprising a value holder that the source object points to~~ wherein the source value holder is an object that wrappers an instance of the target object in order to implement a proxy pattern; and

means for updating a target value holder of the target object to identify the source object, ~~the target value holder comprising a value holder that the target object points to;~~ wherein [[a]] the target value holder is an object that wrappers [[a]] an instance of the target object or source object in order to implement a proxy pattern.